Kernel-based Consensus Clustering for Ontology-embedded Document Repository of Power Substations

L. Yan, W. H. Tang, *Senior member, IEEE,* Q. H. Wu, *Fellow, IEEE,* and J. S. Smith, *Member, IEEE*

***Abstract*—This paper presents a novel consensus clustering (CC) approach for a document repository concerning power substations (PSD) and contributes to the intangible asset management of power systems. A domain ontology model, i.e., substation ontology (SONT), is applied to modify the traditional vector space model (VSM) for document representation, which is concerned with the semantic relationship between terms. A new document representation is generated using a term mutual information matrix with the aid of SONT. In addition, compared with two other novel CC algorithms, i.e., non-negative matrix factorisation-based CC (NNMF-CC) and information theory-based CC (INT-CC), weighted partition via kernel-based CC algorithm (WPK-CC) is utilised to solve the CC issue for PSD. Meanwhile, genetic algorithms (GA) were applied to WPK-CC for PSD, as there are limitations in the original WPK-CC for document clustering. Subsequently, selected mechanisms in each GA's procedure are compared and improved, resulting in comprehensive parameter settings for the PSD CC. Four simulation studies have been designed, in which the results are evaluated by purity validation method and show that the SONT-based document representation and improved WPK-CC, via modified GA, significantly improve the performance of the PSD CC.**

***Index Terms*—power system asset management, document clustering, consensus clustering, ontology, kernel method, GA.**

I. Introduction

ASSET management mainly consists of three types of asset; physical assets, financial assets and intangible assets [1]. As one of the most important intangible assets, digitally stored text knowledge increases very rapidly in volume in power system documentation. Accurate and efficient searching for important technical reports and required academic papers is crucial for power engineers to provide necessary condition monitoring and propose novel strategies for the assessment of equipment (e.g., power transformer) in power systems. A typical document repository (i.e., PSD), which is concerned with power substation related topics only, has been built in our previous research [2]. The PSD contains more than 100,000 text files (all in English), including technical reports, substation maintenance records, published academic papers etc. An ontology-based document search engine was designed and implemented in the PSD, in which the search engine returns a ranked document set with descending relevance scores according to the power engineer's input query. The PSD restricts the searching process to be under the domain of power substation's and the search engine improves both the recall and the precision of searching so that the retrieved documents are highly relevant to the requirements of power engineers. Subsequently, power engineers can provide relevant actions to the power substation. However, a search engine typically returns, to a user, thousands of results based on the query, making it difficult to browse or to identify the relevance.

This paper focuses on document clustering, aiming to examine the existing document repository and divide the repository into several groups in terms of their underlying topics. Briefly, documents with a similar topic have high similarities, while documents within the same cluster are different from documents in other clusters. As an unsupervised learning strategy, clustering has the automated processing capacity for documents without being concerned with the training process, e.g., classification (supervised learning) and annotating the documents manually in advance. In this case, the document repository could be organised into a set of meaningful clusters automatically, which provides an efficient way for power engineers to browse and navigate. This study can be regarded as a logical continuation of work reported in [2].

Document clustering normally contains three fundamental procedures, i.e., document pre-processing, clustering algorithm implementation and result validation [3]. Document representation, which belongs to document pre-processing, always utilises the vector space model (VSM) to represent documents in many existing methods [3], [4]. Basically, VSM is a mathematical model, representing a document as a vector so that a document repository can be expressed by a document-term matrix. The terms refer to the words selected in the repository. This type of representation ignores relationships among important terms that do not co-occur literally. In other words, they only relate documents that use identical terminology [5]. A similar problem also arises in document retrieval in the research field of information retrieval as mentioned in [2]. For instance, documents concerning the topic of transformer diagnosis mainly using the words of "condition assessment" could be clustered into a different group from the documents described by "fault isolation".

To consider the conceptual similarity of terms, much research focuses on adding concepts to the terms or transferring a term vector to a concept vector [5]–[7]. Better results have been achieved, when these approaches are compared with the traditional VSM for selected document datasets. However, these approaches either increase the dimensionality of the text

L. Yan, Q. H. Wu and J. S. Smith are with the Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, L69 3GJ, U.K. W. H. Tang and Q. H. Wu are with the School of Electric Power Engineering, South China University of Technology, Guangzhou, 510641, China. Corresponding author: Professor. W. H. Tang, Email: wenhutang@scut.edu.cn.

data or decrease the amount of information of the raw dataset. They are not practical for solving the large volume document clustering problem.

The document data representation for clustering, in this paper, is inspired by [8], in which a WordNet-based distance measure is proposed. WordNet is utilised as the background knowledge, aiming to generate a "term mutual information matrix". Subsequently, a new data model is obtained by combining the consideration of correlation among terms and the traditional VSM. In this study, a power substation ontology model (SONT) is implemented. SONT is programmed according to the context of power substations only, which is the first ontology model specifically defined regarding the domain of power substations, containing synonyms and hyponyms of each concept. In addition, SONT solves the limitation of WordNet where there is no synonym or hyponym mapped to a power substation concept in some cases. Meanwhile, as an ontology model, SONT is more flexible and expandable with no ambiguity [9].

In practice, different clustering algorithms or a single clustering algorithm with different parameter settings may generate various clustering results [10]. It is not appropriate to decide which clustering result is correct or not, as they are all obtained by equally plausible clustering algorithms [11]. In this case, the method of consensus clustering (CC) is implemented in this study. CC refers to the success of the combination of various clustering solutions, which is a way to improve the performance of any single clustering algorithm. It aims to achieve a comprehensive result with better performance than each single clustering algorithm and the solution should be as similar to all the clustering results [11].

This paper compares three advanced CC algorithms. These algorithms were all originally designed for sample datasets, which have much smaller features than document datasets. In addition, there is no existing comparison study among these CC algorithms. A series of simulation studies were undertaken on selected document datasets with the aim of finding the most advantageous one to handle PSD CC. Results showed that WPK-CC outperformed NNMF-CC and INT-CC on each dataset. Also, GA-embedded WPK-CC was proven to be comparable to the WPK-CC, in which the objective function is solved by the simulated annealing algorithm (SA). More comparisons have been undertaken on the mechanisms of GA.

The proposed approach for PSD clustering consists of PSD representation by the combination of SONT-based VSM and the term mutual information matrix, followed by an appropriate CC algorithm and solved by GA-implemented WPK-CC with optimal schemes of mechanisms and parameter settings. There are four simulation studies designed, in which results show that the proposed approach significantly improves the result for PSD clustering. Meanwhile, with the proposed approach, clustering can be either implemented on the entire PSD and other power system related document repositories or to a retrieved document set so that users only need to browse a small number of accurately retrieved results. As a consequence, this approach improves the efficiency of knowledge acquisition for power engineers or academic staff and contributes to power system asset management.

II. Background Knowledge

A. Document Data Pre-processing

Document data pre-processing normally consists of tokenisation, linguistics and then representing it as a mathematical model [12]. Tokenisation aims to transform the content of a document into a sequence of terms, eliminates the punctuation and performs common stop words removal (e.g., "a", "an", "and", "in" etc. are removed). There are a large number of stop words in every document, which is not helpful for searching. For the linguistics, all the terms are changed to lowercase and the common morphological and inflectional forms are eliminated by a suffix stripping algorithm [12], including stemming (e.g., automated $\rightarrow$ automate) and lemmatisation (e.g., criteria $\rightarrow$ criterion). Subsequently, assume $D = \{d_1, d_2, ..., d_n\}$ denotes a set of "n" documents, where $v = 1, 2, ..., n$ and each $d_v \in D$ is a tuple of some m-dimensional space. In traditional VSM, each document d_v can be represented by $t_{vu} = \{t_{v1}, t_{v2}, ..., t_{vm}\}$ or $\{t_1, t_2, ...t_m\}$ (as terms in the vocabulary extracted from the corresponding document collection are fixed). The corresponding term weight is denoted by $\omega_{vu} = \{\omega_{v1}, \omega_{v2}, ..., \omega_{vm}\}$. The weight can either be the term frequency, i.e., $tf_v = \{tf_{v1}, tf_{v2}, ..., tf_{vm}\}$ or based on term frequency-inverse document frequency (tf-idf) [13].

B. Document Clustering Algorithms

1) Single Clustering Algorithm: K-means is one of the best-known exclusive clustering algorithms, performing as a foundation algorithm for researchers to design new clustering algorithms [10]. It is an iterative procedure that is guaranteed to converge, though not always to the best solution, and "k" refers to the number of clusters. The obtained cluster set is denoted by $C = \{C_1, C_2, ..., C_3\}$. K-means revolves around the placement and replacement of "k" centroids. The centroid of a cluster is defined to be the average of the vectors in that cluster, Each data point is put in the cluster associated with the nearest centroid. The algorithm aims at minimising the objective function as shown in (1).

$$J = \arg\min_{C} \sum_{i=1}^{k} \sum_{d \in C_i} \|x - \mu_i\|^2, \tag{1}$$

where μ_i is the mean of points in C_i. The $\|\cdot\|^2$ denotes the chosen distance measurement (e.g., the Euclidean distance) between a data point and the cluster centre.

2) Consensus Clustering: Given a set of documents, CC consists of two steps; generation and consensus function. The generation step aims to obtain a set of alternative clustering results named partitions. The consensus function is applied to combine these partitions. For instance, there is a document repository containing nine documents, $\{d_1, d_2, ..., d_9\}$ with the underlying classification "111222333" and k-means with different initialisations generates ten different partitions, e.g., "111122233" (In this case, nine documents are grouped into three clusters, and four documents belong to the cluster 1) , "112223233" and etc. If a consensus function is applied, it is

expected to obtain a solution with the underlying classification, i.e., "111222333". Suppose that s partitions, i.e., $P_D = \{P_1, P_2, ..., P_s\}$, are obtained. Each partition P_l, $l = 1, ..., s$ consists of a set of clusters $C^l = \{C_1^l, C_2^l, ..., C_k^l\}$ where k is the number of clusters for partition P_l and $X = \bigcup_{c=1}^{k} C_c^l$. The consensus partition P^* is obtained by the solution of an optimisation problem, as illustrated by (2) [14]:

$$P^* = \arg\max_{P \in P_D} \sum_{j=1}^{s} k(P, P_j), \tag{2}$$

where k is a similarity measure between two partitions. Alternatively, it can also be explained and denoted by minimising the dissimilarities within P_D.

3) Consensus Functions:

- WPK-CC was proposed by Vega-Pons and Correa-Morris [14]. This algorithm involves partition relevance analysis. Typically, each partition is validated by some property validity indexes (PVI) or internal validation methods and assigned by a set of weight (ω_l) based on the entropy. The similarity measure between partitions ($k(P_i, P_j)$), which is proven to be positive semidefinite, is based on analysing each subset of D. In this case, k is a kernel function and there exists a map from P_D into a Hilbert space $\mathcal{H}$ such that $k(P_i, P_j) = \langle \phi(P_i), \phi(P_j) \rangle_{\mathcal{H}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the dot product in the Hilbert Space $\mathcal{H}$ [15]. Subsequently, the objective function is transferred to the equivalent equation in the Reproducing kernel Hilbert space. The approximate solution $\widehat{P}$ is defined by (3).

$$\hat{P} = \arg\min_{P \in P_D} \left\| \widetilde{\phi}(P) - \widetilde{\phi}(P^*) \right\|_{\mathcal{H}}^2, \tag{3}$$

with

$$\left\| \widetilde{\phi}(P) - \widetilde{\phi}(P^*) \right\|_{\mathcal{H}}^2 = \widetilde{k}(P, P) - 2 \sum_{l=1}^{s} \widetilde{\omega}_l \widetilde{k}(P, P_l) + \sum_{i=1}^{s} \sum_{j=1}^{s} \widetilde{\omega}_i \widetilde{\omega}_j \widetilde{k}(P_i, P_j).$$

where $\widetilde{\phi}(P)$, $\widetilde{\omega}$ and $\widetilde{k}$ are normalised $\phi(P)$, ω and k, respectively.

- NNMF-CC was proposed by Li and Ding, which is based on non-negative matrix factorization (NNMF) referring to the problem of factorising a given non-negative matrix into two matrix factors [16]. This algorithm starts from defining a connectivity matrix, which is used to demonstrate the relationship between the element-wise distance in two partitions. Thus, a primary objective function is obtained. Afterwards, cluster indicators are designed according to the connectivity matrix so that the objective function is transferred into a symmetric NNMF and solved by NNMF multiplicative update rules.
- INT-CC aimed to minimise an information theoretical criterion function using GA, which was proposed by Luo and Jing [17]. This method uses a metric between partitions based on the entropy between partitions and the objective function is solved by GA.

C. Validation Methods

Purity refers to an external validation method, which is the percentage of the total number of objects that were correctly clustered [4]. Thus, a clustering result with a higher purity indicates it is an optimal result. The purity of a single cluster C_j is defined as the fraction of objects in the cluster that belong to the dominant class contained within that cluster:

$$P(C'_i, C_j) = \frac{1}{n_j} \max_i \{N_{ij}\}$$

where N_{ij} denotes the size of the intersection $\left|C_i' \cap C_j\right|$ between the class C_i' and cluster C_j; n_j is the number of data in cluster C_j. The overall purity of a cluster is defined as the sum of the individual cluster purities, weighted by the size of each cluster, which is illustrated in (4).

$$P(C', C) = \sum_{j=1}^{k} \frac{n_j}{n} P(C', C_j) \tag{4}$$

III. Proposed Methods

A. PSD Modification with Ontology

1) A Domain Ontology SONT: SONT has been developed, in our previous research [2], using the Protégé ontology development software [18], with the aim of expanding the original query for a document search. The relationship among the related terms are described with web ontology language (OWL) semantics. Considering the domain and scope of SONT, "Power System" is defined as "Thing" at the top ontology level, and a top-down development process is utilised to define the corresponding classes and the class hierarchy in the Protégé. The second level classes contain nearly all the important aspects of power substations, such as "Action", "Attributes", "Device", "Other assets", "Status" and "Units", etc. For instance, "Action" consists of four subsequent subclasses; "Monitoring", "Restoration", "Protection", and "Vibration". The current version of SONT has 413 classes, 67 properties and 31,579 instances. More details of SONT can be found in [2].

2) PSD with SONT-based VSM and Term Mutual Information: From the linguistics point of view, some researchers have verified that there exist mutual relations between the terms in a document-term vector [19]. Therefore, it is essential to take these term relationships into consideration rather than simply using the traditional VSM [20].

The first step of this method is to examine whether a term t_{u_1} is semantically correlated to the other term t_{u_2} with SONT. In other words, this step aims to check the synonym and hyponym set of each term, and an indicator, i.e., $\delta_{u_1 u_2}$, is defined to present the semantic information between two terms. If t_{u_2} is a synonym or hyponym of t_{u_1}, $\delta_{u_1 u_2}$ will be set to a coefficient, otherwise, $\delta_{u_1 u_2}$ will be set to zero. Thus, with the $\delta_{u_1 u_2}$ embedded, the term frequency can be modified by (5).

$$\widetilde{\omega}_{v u_1} = \omega_{v u_1} + \sum_{\substack{u_2 = 1 \\ u_2 \neq u_1}}^{m} \delta_{u_1 u_2} \omega_{v u_2}. \tag{5}$$

(a) Traditional VSM

term	d_1	d_2	d_3
power	5	8	6
transformer	10	12	10
fault	5	2	0
diagnosis	10	0	0
detection	0	8	0
assessment	0	6	0
magnetic	0	0	5
circuit	0	0	5
optimisation	0	0	8

$\rightarrow$

(b) SONT-based VSM

term	d_1	d_2	d_3
power	5	8	6
transformer	10	12	10
fault	5	2	0
diagnosis	10	11.2	0
detection	8	12.8	0
assessment	8	12.4	0
magnetic	0	0	5
circuit	0	0	5
optimisation	0	0	8

TABLE I
THE COMPARISON BETWEEN TRADITIONAL VSM AND SONT-BASED VSM

An example has been illustrated as follows: there are three documents, i.e., $\{d_1, d_2, d_3\}$. d_1 and d_2 mainly introduce the topic of transformer fault diagnosis, while d_3 covers the power transformer design. The term frequency, which is based on the traditional VSM, is given by Table I (a). Diagnosis, assessment and detection in the power substations discipline are semantically related to each other. According to (5) with $\delta_{u_1u_2} = 0.8$, the traditional VSM can be transformed into the SONT-based VSM as illustrated in Table I (b). The Euclidean distance, based on the traditional VSM between two documents, is $dis(d_1, d_2) = \sqrt{(d_1 - d_2)(d_1 - d_2)^T} = \sqrt{\sum_{u=1}^{m}(\omega_{1u} - \omega_{2u})^2}$. If the term frequency (tf_{vu}) is chosen to be the term weight, the distance between d_1 and d_2, d_2 and d_3, d_1 and d_3 are 14.8997, 15.0333 and 15.4919, respectively. In the SONT-based VSM, the above distances are updated to 8.1142, 23.8546 and 19.1833, respectively. The influence of involving the semantic relations for VSM is remarkable, as the distance between d_1 and d_2 decreases and the distance between d_1 and d_3, d_2 and d_3 increases significantly. As a consequence, d_1 and d_2 have more chances to be clustered into the same group in the cluster analysis, and d_3 will be assigned to another cluster.

Although PSD contains different topics and the documents can be divided into several categories in terms of these topics, they are all under the context of power substation. According to the Harris distributional hypothesis, the words or terms occurring in the same contexts tend to have similar meanings [21]. Thus, there exist some syntactic surface dependencies between a pair of terms that simultaneously occur in the repository. The syntactic surface dependencies are defined as term mutual information (TMI) and computed by the cosine similarity. The TMI between term t_1 and term t_2 is illustrated in (6), in which the similarity of each pair of terms, i.e., mutual information matrix (MIM), in a given repository can be computed by (7).

$$TMI_{t_1t_2} = \frac{\sum_{v=1}^{n} \widetilde{\omega}_{v1}\widetilde{\omega}_{v2}}{\sqrt{\sum_{v=1}^{n}\widetilde{\omega}_{v1}^2} \cdot \sqrt{\sum_{v=1}^{n}\widetilde{\omega}_{v2}^2}}. \tag{6}$$

$$MIM = \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1u} & \dots & \sigma_{1m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{v1} & \dots & \sigma_{vi} & \dots & \sigma_{vm} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{m1} & \dots & \sigma_{mu} & \dots & \sigma_{mm} \end{bmatrix}, \tag{7}$$

where σ_{vu} denotes the mutual similarity between term t_v and t_u. It can be noted that the similarity of (t_v, t_u) is equivalent to (t_v, t_u). Thus, MIM is symmetric. In addition, there is no difference between the same term, i.e., (t_u, t_u). Therefore, MIM can be modified, which is illustrated by (8).

$$M = \begin{bmatrix} 1 & \dots & \sigma_{u1} & \dots & \sigma_{v1} & \dots & \sigma_{m1} \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \sigma_{u1} & \dots & 1 & \dots & \sigma_{vu} & \dots & \sigma_{mu} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \sigma_{v1} & \dots & \sigma_{vu} & \dots & 1 & \dots & \sigma_{mv} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \dots & \sigma_{mu} & \dots & \sigma_{mv} & \dots & 1 \end{bmatrix} \tag{8}$$

The elements in M are all greater than or equal to zero so that M is symmetric positive semidefinite [22]. The mutual information matrix can be decomposed by an orthogonal matrix A and a diagonal matrix D, as presented by (9).

$$M = ADA^T = A\sqrt{D}\sqrt{D}A^T = (A\sqrt{D})(A\sqrt{D})^T = BB^T, \tag{9}$$

where B refers to the correlation factor matrix, and $B = A\sqrt{D}$. According to the Euclidean distance based on the term frequency between two documents, the distance with the term mutual information matrix can be denoted by (10).

$$md(d_1, d_2) = \sqrt{(d_1 - d_2)M(d_1 - d_2)^T}. \tag{10}$$

This distance measure refers to the Mahalanobis distance, where the matrix M is defined as the dimensions correlation coefficient appearing in the Mahalanobis distance [23]. It is noticed that (10) turns into a Euclidean distance, if M is an identity matrix. According to (9), the distance $md(d_1, d_2)$ can be modified as follows:

$$\begin{aligned} md(d_1, d_2) =& \sqrt{(d_1 - d_2)M(d_1 - d_2)^T} \\ =& \sqrt{(d_1 - d_2)BB^T(d_1 - d_2)^T}, \\ =& \sqrt{(\widehat{d_1} - \widehat{d_2})(\widehat{d_1} - \widehat{d_2})^T} \end{aligned} \tag{11}$$

where $\widehat{d_1} = d_1B$ and $\widehat{d_2} = d_2B$. Thus, the Mahalanobis distance between d_1 and d_2 is equivalent to the Euclidean distance between $\widehat{d_1}$ and $\widehat{d_2}$.

B. WPK-CC with Genetic Algorithm

Each partition is an integer string with fixed length, of which each component represents a cluster label for one document. Simulated annealing (SA) has been applied to solve the consensus function of WPK-CC. In each iteration, SA creates a new solution by changing only one label to another. In this case, SA in WPK-CC is similar to GA, when the *PopSize* (*PopSize*) of GA is one and only mutation operates without crossover. GA returns a set of solutions rather than a single solution, permitting more chances to obtain a more approximate solution. In addition, if the integer string is very long (e.g., more than 100,000 bits), WPK-CC will converge extremely slowly. Moreover, the parameters in SA, i.e., initial temperature and the temperature decreasing rate, are more

TABLE II
DOCUMENT REPOSITORIES

Dataset	Description	n	m	classes	documents per class
bbcsports[1]	Sports news articles from BBC	737	4,613	5	101-124-265-147-100
bbc[2]	Articles from BBC	2,225	9,635	5	510-386-417-511-401
TDT2-6[3]	Subset of TDT2	6,523	36,771	6	1844-1828-1222-811-411-407
PSD	Power substation document corpus	136,735	700,083	6	41223-21482-32115–10101-22583-9231

likely to be determined empirically. It is difficult to find a balanced temperature decreasing rate in SA, which concerns both accuracy and efficiency. GA contains various mechanisms and parameters in each step (e.g., selection, crossover and mutation), which allows a more thorough optimisation for a CC application. Meanwhile, utilising GA to deal with the clustering problem is more straightforward compared to other algorithms, as it is not necessary to involve encoding and decoding for the chromosomes. It can be concluded that GA is a more advantageous method to solve the consensus function than SA or other optimisation methods. Also, it is noticed that a selected mutation point can mutate to any cluster number. In this paper, if a mutation point is chosen, all alleles in the population will be analysed. In general, if a cluster number occurs more frequently than others, it will have a higher probability that other cluster numbers could be mutated to. For instance, if mutation occurs at the second gene, i.e., "**1**", of a chromosome "1**1**1222233344" (the length denotes the number of documents is 12) and a vector of its alleles is "111223" (the length denotes *PopSize*$= 6$), the potential mutated gene can be "2", "3" or "4". Without considering the original gene "1, the probabilities to mutate to "2", "3" and "4" are $2/3$, $1/3$, and 0, respectively. In order to consider every possible number, Baker's linear ranking (LR) [24] is used. The probabilities are changed to $1/2$, $1/3$, $1/6$, respectively. This mutation is designed especially for document clustering, and the authors define it as DC-mutation.

IV. SIMULATION STUDIES

In this paper, four case studies are applied to PSD. Briefly, case 1 aims to compare the three CC algorithms as well as k-means on different text datasets so that the most advantageous CC algorithm for document clustering can be selected. Case 2 discusses PSD clustering with background knowledge embedded. Case 3 compares the original WPK-CC with SA (or WPKSA) and GA-embedded WPK-CC (or WPKGA). Case 4 analyses the impact of GA's mechanism to PSD CC. Typically, five values of *PopSize* (i.e., 10, 20, 50, 100 and 200), four selection schemes (i.e., roulette wheel (RW), LR, elitism with roulette wheel (ERW) and elitism with LR (ELR)), four crossover schemes (i.e., one-point, two-points, uniform and binominal crossover) and three mutation schemes (bit-flap, adaptive mutation and DC-mutation) are compared. In addition, the crossover rate and mutation rate are also selected by comparison results. Each simulation study starts from generating 20 partitions by k-means with random initialisations. The initial settings follow the original papers [14] [16] [17] and four PVIs have been selected; variance, connectivity, silhouette width and Dunn index [25] [26]. Purity is the validation method considered in these case studies and it is the average value of 10 independent algorithm runs. Fig. 1 illustrates the overall flowchart for each simulation study and the numbers by each arrow denote the relevant cases.

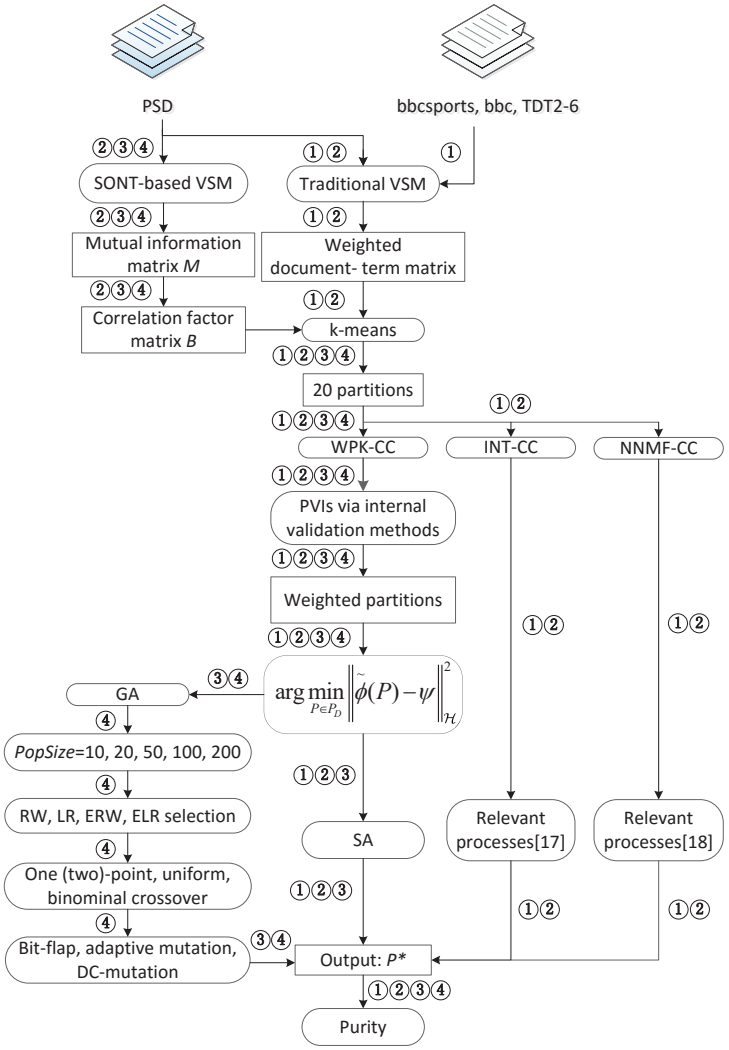

Fig. 1. The flowchart of simulation studies. ①, ②, ③ and ④ represent case study 1, 2, 3 and 4, respectively

A. Case study 1:

This case study examines the impact of three CC algorithms, which were tested on four selected text datasets. The information of each dataset is shown in Table II. The results are shown in Table III. It can be concluded that all the CC algorithms have significant improvements compared with the single clustering algorithm. Among them, WPK-CC outperforms NNMF-CC and INT-CC in all purity levels. These can be explained because the WPK-CC involves property

[1]Available from http://mlg.ucd.ie/datasets/bbc.html
[2]Available from http://mlg.ucd.ie/datasets/bbc.html
[3]Available from http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

validity indexes to each partition. As some of the partitions from the generation step might be wrongly generated, which can be regarded as noise partitions. The partition relevance analysis is a method to assign a small weight to these noise partitions. It avoids a simple average of the set of partitions producing a worse clustering result. The results demonstrate that NNMF-CC and INT-CC are competitive in text datasets.

TABLE III
PURITY (%) OF DIFFERENT CLUSTERING ALGORITHMS ON DIFFERENT DOCUMENT DATASETS

	k-means	NNMF-CC	WPK-CC	INT-CC
bbcsports	43.31	45.33	51.20	45.61
bbc	27.88	38.98	41.51	40.29
TDT2-6	42.13	51.24	53.39	49.23
PSD	31.72	35.29	38.43	33.19

B. Case study 2:

This case study focuses on the influence of involving the background knowledge to the document dataset, i.e., PSD with SONT-based VSM, which is named as modified PSD and denoted by MPSD. The Euclidean distance measure between two documents are transformed to a Mahalanobis distance, which takes the correlation between each pair of terms into account. It is more reasonable to involve the term mutual information than to ignore the inter-relations between terms. According to (1) and (11), the distance between a document and its cluster centroid is denoted by (12).

$$\begin{aligned} md(d_j, C_l) &= \sqrt{(d_j B - C_l B)(d_j B - C_l B)^T} \\ &= \sqrt{(\widehat{d}_j - \widehat{C}_l)(\widehat{d}_j - \widehat{C}_l)^T}, \end{aligned} \tag{12}$$

where $\widehat{d_j}$ is the transferred document vector and $\hat{C}_l$ is the l_{th}cluster's centroid. Thus, the Euclidean distance between a document and its cluster centroid is defined by (13).

$$d(\widehat{d}_j, \widehat{C}_l) = \sqrt{(\widehat{d}_j - \widehat{C}_l)(\widehat{d}_j - \widehat{C}_l)^T}. \tag{13}$$

The standard k-means can be applied to the MPSD. Results are shown in Table IV. WPK-CC outperforms the other clustering algorithms. With SONT embedded PSD, each algorithm reaches an improved purity. It should also be noted that the purities of the MPSD with CC algorithms have more significant improvement than that with average k-means. This may be due to that the noise partition or a wrongly generated partition cannot be completely avoided during the generation step. In this case, compared with PSD, the growth rate of using MPSD may not be significant, when using the standard k-means. However, PSD contains 136,735 documents, which means that even an improvement of 0.33% for MPSD with k-means assigns more than 400 documents correctly into the underlying clusters compared with the normal PSD without considering the relationship between terms. From this point, each CC algorithm has improved performance, as the improvement of the best performing WPK-CC with MPSD is 1.72%. Thus, more than 2300 documents are correctly clustered compared with WPK-CC with the single PSD.

Also, the clustering results have been improved by involving the term mutual information. SONT is implemented to add background information to the original dataset, i.e., PSD. As a consequence, there are more relevant documents being assigned into the same cluster so that the accuracy of the cluster result is improved significantly.

TABLE IV
PURITY (%) OF DIFFERENT CLUSTERING ALGORITHMS ON PSD AND MPSD

	k-means	NNMF-CC	WPK-CC	INT-CC
PSD	31.72	35.29	38.43	33.19
MPSD	32.05	37.28	40.15	34.66
Increment (%)	0.33	1.99	1.72	1.47

C. Case study 3:

This case study aims to compare WPKSA and WPKGA. MPSD is the dataset representation in the rest of the case studies. It is also noted that WPKSA aims to find the minima of the objective function as shown in (3). It is noted that the first and third term of (3) are fixed numbers, as $\tilde{k}(P, P)$ denotes the similarity measure between an intermediate solution partition and itself, and $\sum_{i=1}^{s}\sum_{j=1}^{s}\tilde{\omega}_i\tilde{\omega}_j\tilde{k}\left(P_i, P_j\right)$ denotes the sum of weighted similarity measures among 20 partitions. The objective function or fitness function of WPKGA aims to find the maxima of the second term of (3), i.e., $2\sum_{i=1}^{s}\tilde{\omega}_i\tilde{k}\left(P, P_i\right)$. In any case, both WPKSA and WPKGA are applied to seek the optima of the objective function and the clustering results are evaluated by purity.

In addition, in the previous cases, each CC algorithm is terminated by a fixed number of iteration steps. As stochastic algorithms, SA and GA involve iterative processes before obtaining the results and the termination condition is not guaranteed to be known. In this case, the termination statuses for WPKSA and WPKGA were modified in order to consider the nature of the iterations and are presented in Table V. The iteration of WPKSA terminates when it reaches the pre-defined maximum generations (i.e., $IMax$) or the objective function $\widehat{P} = 0$ becomes zero. WPKGA terminates when either the iteration reaches $IMax$ or the difference between the best objective value F_{best} and the corresponding average fitness or purity in the population, i.e., F_{avg}, is not significant.

TABLE V
TERMINATION STATUS OF WPKSA AND WPKGA

	Termination status	
WPKSA	$IMax = 10000$	$\widehat{P} = 0$
WPKGA		$\frac{\lvert F_{\text{avg}} - F_{\text{best}}\rvert}{\lvert F_{\text{avg}}\rvert} \leq \varepsilon$

Fig. 2 illustrates the generations against purity of WPKSA and WPKGA on MPSD. Both the average and maximum purity for each iteration of WPKGA are presented. WPKSA starts the iteration from a higher purity (more than 25.88%) and WPKGAs start from an apparently lower purity (less than 15%). The states in WPKSA are partitions, and the idea is to start from an initial partition, which is the partition with best performance, through an iterative process, and to obtain a very close partition to the consensus one. On the other hand, the initial population of WPKGA is randomly generated. Therefore, the difference between WPKSA and WPKGA in Fig. 2 at the starting point of the iteration is predicted. In

addition, WPKGA converges at less than 8000 generations, and WPKSA is terminated by $IMax$. As previously mentioned, the similarity measure between partitions in WPKSA is based on the intersection of objects. If the size of a dataset is too large, each iteration will generate very little improvement, resulting in an extremely slow convergence speed. In contrast, WPKGA operates with a population of chromosomes, and the selection, crossover and mutation mechanisms of GA provide more chances to achieve a faster convergence speed.

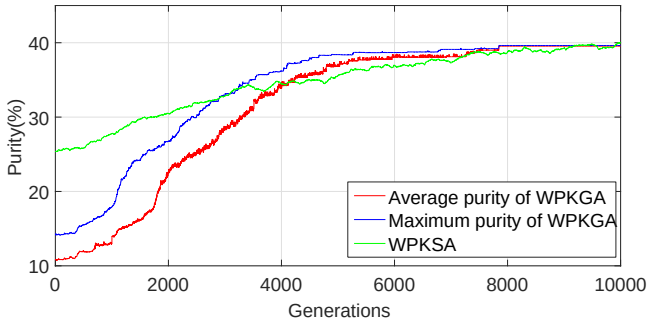

Fig. 2. The comparison among the average purity of WPKGA of the entire population, maximum purity of WPKGA in the population and the purity of WPKSA

Table VI illustrates the result of WPKSA and WPKGA implemented on MPSD. The "max" and "average" denote the maximum and average purity of 50 resultant chromosomes in each generation of WPKGA, respectively. Both of the final maximum purity (39.63%) and average purity (39.60%) of the resultant GA population perform better than that using NNMF-CC (37.28%) and INT-CC (34.66%), and the maximum purity (39.63%) is slightly smaller than the result of WPKSA (40.15%). It indicates that the result of WPKGA on MPSD is more comparable to WPKSA, when compared with NNMF-CC and INT-CC. Also, the convergence speed of WPKGA is significantly slower than WPKSA. In practice, the best performing chromosome from the last population is always selected as the solution of the optimisation. Therefore, in case study 4, only the best chromosome in the final population is taken as the optima.

TABLE VI
PURITY OF OPTIMUM SOLUTION FOR WPKSA AND WPKGA ON MPSD

	WPKSA	WPKGA	
		max	average
Purity (%)	40.15	39.63	39.60
Termination condition	$IMax$	7847	7847

D. Case study 4:

This case study aims to evaluate the performance of GA in WPKGA with different mechanisms and parameter settings, and implemented to solve MPSD CC. Each comparison is carried out based on the optimal settings in the previous GA's procedure. Fig. 3 illustrates the generation of WPKGA with different *PopSize* and the results are presented in Table VII. For *PopSize* of $10, 20, 50$ and 100, the results show that the increase of the *PopSize* significantly improves the purity of the clustering result, while the convergence speed decreases. In the previous case studies, *PopSize* was 50, which produces a much better result (39.63%) compared with *PopSize* of 10 (32.23%) and 20 (36.56%). When *PopSize* increases to 100,

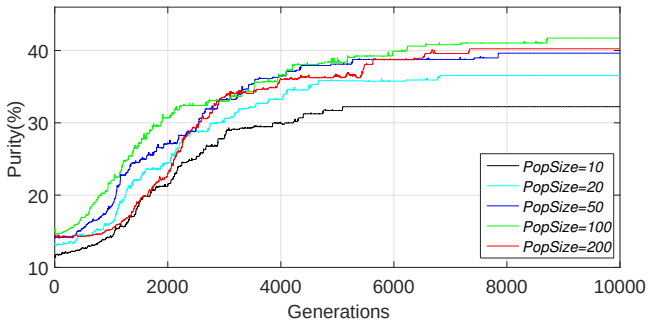

Fig. 3. WPKGA for MPSD clustering with different *PopSize*

it achieves a significant higher purity (41.72%) and slower convergence speed (8712). When *PopSize* is 200, WPKGA is terminated by $IMax$ and the purity (40.24%) is worse than *PopSize* of 100. In addition, when *PopSize* changes from 10 to 50, significant improvements of the purity are obtained. In contrast, when *PopSize* increases from 50 to 200, the improvement of the purity is less significant. It is concluded that if *PopSize* is too small, population diversity become very low. In this case, each new generated population has little chance to perform crossover or mutation operations, resulting in premature termination of the algorithm. On the contrary, if *PopSize* is too large, the fitness level doesn't always increase and may actually reduce. Thus, *PopSize* $= 100$ was selected to be an optimal setting for the first step of WPKGA on MPSD.

TABLE VII
PURITY AND CONVERGENCE OF WPKGA WITH DIFFERENT *PopSize* ON MPSD

	10	20	50	100	200
Purity (%)	32.23	36.56	39.63	41.72	40.24
Termination Condition	5096	6819	7847	8712	$IMax$

In order to improve the readability, make it easier for analysis and to indicate the statistical significance, error bars are used to present the results for the remaining tests. The aim of using error bars in presenting the CC results is to show the average purity and generations of each GA mechanism and parameter setting in 10 WPKGA runs on MPSD in a clear way. In addition, the standard deviation (SD) of each 10 purities is also shown in the same figure so that the stabilisation of each mechanism can be evaluated at the same time. Fig. 4 shows the generation of WPKGA for MPSD CC with four selection mechanisms; ERW, ELR, RW and LR, when *PopSize* was 100. Each blue bar stands for one mechanism or one GA parameter setting. The "Y" axis represents the purity and the "X" axis shows the generations. The top ends of the blue bars denote the average purities of 10 WPKGA runs, locating on the convergence iterations, and the red line segments represent the SD of each 10 WPKGA runs. U and L denote the upper SD and lower SD, respectively. The SD quantifies by how much the values vary from one another. A long red line segment shows that the concentration of the values that the average was calculated on is low, and thus the average value is uncertain. Conversely, a short line segment means that the concentration of values is high and that the average value is more certain.

Results show that WPKGA with ERW (7187) terminates faster

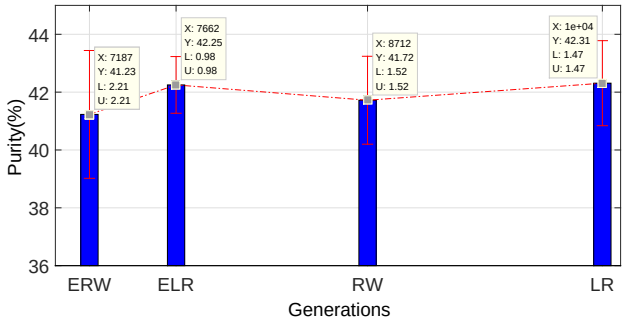

Fig. 4. WPKGA for MPSD clustering with different selection mechanisms and *PopSize* = 100, where "X" represents the generation at convergence for each mechanism; "Y" shows the average purity; "L" & "U" denote the lower and upper standard deviations

than other selection mechanisms and the LR has the slowest convergence speed, which is terminated by $IMax$. As ERW keeps and directly copies the best 2% chromosomes to the next generation without crossover and mutation operations, it avoids the disruption of the best chromosomes. The LR has the highest purity (42.31%), which is comparable to ELR (42.25%). Both of these two mechanisms perform better than RW (41.72%) and ERW (41.23%) has the smallest purity among the four selection types. The SD of ERW (2.21%) is larger than the others, which means the result of WPKGA with ERW is less stable than the others. LR overcomes the limitation of RW, which is if the best chromosome has a much better fitness, other chromosomes will have fewer chances to be selected. Although LR sacrifices the convergence speed, it keeps the diversity of population and avoids a local minimum. ELR combines the advantages of LR and ERW, producing a good CC result with an acceptable termination status. Therefore, ELR was identified as the optimal mechanism for the simulations.

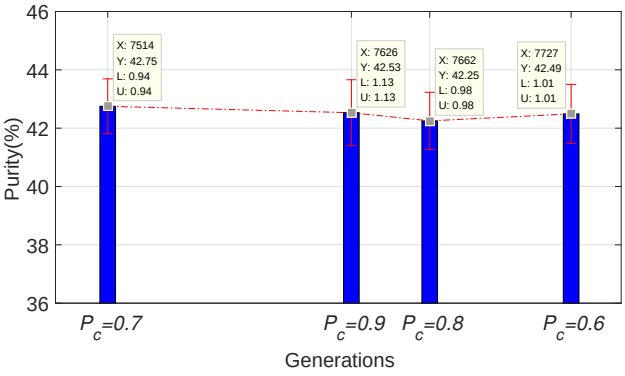

Fig. 5. WPKGA for MPSD clustering with different crossover rates when 1. *PopSize* = 100; 2. selection type is ELR

Fig. 5 illustrates the impact of different crossover rates (P_c), i.e., $0.6, 0.7, 0.8$ and 0.9. Generally, the crossover rate should be high (e.g., 0.9) so that most individuals can be involved in the genetic process. Results show that at $P_c = 0.7$, the purity obtained is the best (42.75%), and it converges faster than other crossover rates with the smallest SD. The purity and SD of $P_c = 0.6$ and $P_c = 0.9$ are similar, i.e., (42.49%) and (1.01%); (42.53%) and (1.13%), respectively. However, $P_c = 0.6$ has a slower convergence speed (7727) than $P_c = 0.9$ (7626). Although $P_c = 0.8$ produces a worse purity than the other crossover rates, the stability (SD= 0.98%) is better than $P_c = 0.6$ and $P_c = 0.9$. It can be concluded that a high crossover rate is not always the best, as it also depends on the specific problem or other parameter settings. Thus, for the remaining results, the crossover rate was set to be $P_c = 0.7$.

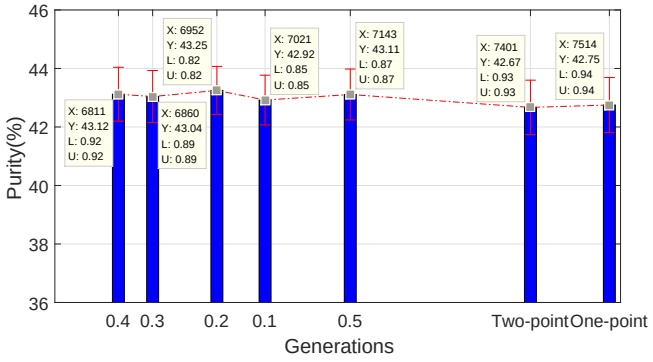

Fig. 6. WPKGA for MPSD clustering with different crossover types when 1. *PopSize* = 100; 2. ELR selection; 3. P_c is 0.7, numeric values on the "X" axis represent the probability of binominal crossover

Fig. 6 presents different crossover types, i.e., one-point crossover, two-point crossover, binominal crossover with probability (P) of 0.1, 0.2, 0.3, 0.4 and 0.5. When P equals to 0.5 in binominal crossover, it also refers to the uniform crossover. Here, it is not necessary to consider the cases, where $P > 0.5$, due to the symmetry of binominal crossover. The results illustrate that one-point crossover has the worst performance on purity (42.75%), convergence speed (7514) and SD (0.94%) and two-point crossover performs only slightly better than one-point, i.e., purity (42.67%), convergence speed (7401) and SD (0.93%). Binominal crossovers with all probabilities have competitive purities. Among them, the uniform crossover has the smallest purity (42.92%) with an average SD (0.85%) and binominal crossover with $P = 0.1$ converges slower (7143) than the other binominal crossover probabilities. The purity of $P = 0.2$ (43.25%) is the best performing binominal crossover with the smallest SD (0.82%) and the convergence speed (6952) is only slower than $P = 0.3$ (6811) and $P = 0.4$ (6860). Binominal crossovers allow the offspring chromosomes to search all possibilities of recombining those different genes in their parents. Considering all the aspects discussed above, binominal crossover with probability of 0.2 was selected as the optimal crossover mechanism.

Fig. 7 shows WPKGA for MPSD clustering with different mutation rates (P_m), i.e., $0.05, 0.1, 0.15$ and 0.2. Among all the mutation rates, $P_m = 0.15$ outperforms the other rates on purity (43.96%) with a middle level of SD (0.93%). Although $P_m = 0.05$ converges faster than other values of P_m, it has a distinct large SD (1.31%) and a low purity (41.38%), which is only slightly larger than *PopSize*= 50 in Fig. 3 without any advanced settings. $P_m = 0.2$ reaches the convergence status slower than the others (7321). Results show that mutation rate cannot be either too high or too low. If P_m is set too high, the search will turn into a primitive random search. If P_m is too low, the diversity of population cannot be ensured. As it is difficult to ensure the correct value for P_m, adaptive mutation was implemented. In each iteration, the mutation rate is reset

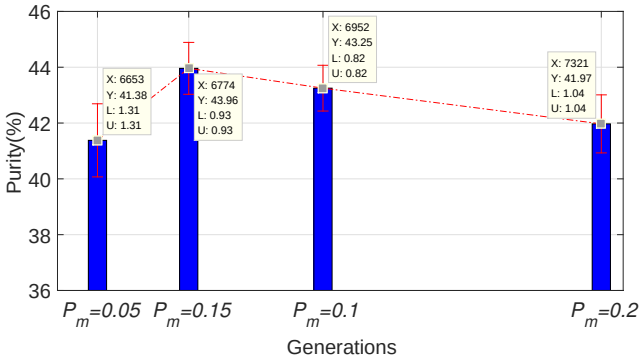

Fig. 7. WPKGA for MPSD clustering with different mutation rates when 1. the *PopSize* $= 100$; 2. ELR selection; 3. P_{c} is 0.7 with a binominal crossover probability of 0.2

automatically depending on the property of the population. Only the maximum and minimum mutation rates need to predefined.

Fig. 8 compares different mutation types. $P_{\mathrm{m}} = 0.15$ was set to bit-flip mutation. For adaptive mutation, the P_{m} range was set to be $[0.05, 0.2]$ and followed by the strategy in [27]. DC-mutation represents the mutation mechanism for document clustering based on adaptive mutation. Results show that DC-mutation outperforms adaptive mutation and bit-flip mutation on each aspect, i.e., purity (45.74%), convergence speed (6598) and SD (0.81%). Among them, bit-flip mutation performs the worst with purity (43.96%), convergence speed (6774) and SD (0.93%).

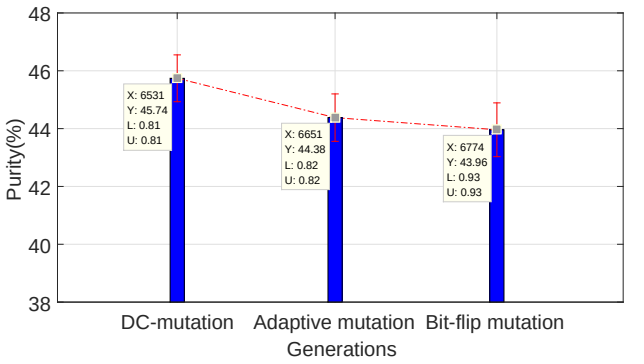

Fig. 8. WPKGA for MPSD clustering with different mutation types when 1. *PopSize* $= 100$; 2. ELR selection; 3. crossover rate is 0.7 with a binominal crossover probability of 0.2; 4. P_{m} is 0.15; 5. the range of adaptive mutation rate is $[0.05, 0.2]$

DC-mutation is specially designed for document clustering. Since the chromosomes for document clustering are not based on binary coding, the selected point is able to mutate to any other integers based on the cluster number. The GA is classed as converged, when the difference between the average performance and the maximum performed chromosome is not significant. In other words, most chromosomes tend to be the same as possible (the Hamming distances [28] amongst the chromosomes decrease) with generations. Therefore, the alleles of each chromosome are analysed to produce a vector with the same length of the population. The cluster number with the most occurrences may not be the underlying cluster label, which can be mutated to. However, it is reasonable to assign it with a higher probability. In order to guarantee that the selected gene can mutate to any cluster label, LR is implemented. It avoids the cluster label with the most occurrences which has much more opportunity to be mutated to and assigns a small probability to the absent cluster label in the allele set as illustrated in the example mentioned in Section III-B.

V. Conclusion

This paper proposes an improved PSD clustering method concerning three aspects, i.e., background knowledge involved PSD pre-processing, an advantageous CC algorithm according to the relevant comparison and significant improvements to the original WPK-CC with GA. Firstly, three CC algorithms and k-means have been applied to selected document datasets and the results show that WPK-CC outperforms NNMF-CC and INT-CC. An ontology model has been applied to add background knowledge to PSD, concerning the semantic correlation among terms. Subsequently, the original PSD was modified by the term mutual information and the correlation factor matrix was obtained. Thus, the modified PSD enhances the performances of each clustering algorithm.

Meanwhile, WPKGA has been compared with the original WPKSA. GA was proven to be more suitable to handle document clustering issues. As GA contains different mechanisms and parameters in each step, comparisons of these mechanisms have been evaluated and discussed. It is concluded that PSD clustering has been significantly improved, providing power engineers a more accurate and convenient way for important text files mining. In this case, clustered PSD improves the efficiency of relevant document browsing and navigation, benefiting the power system asset management.

References

[1] M.Beardow. *Economics of asset management: drawing it together*. ESAA 2003 Residential school in Electrical power Engineering, 2003.

[2] W.H Tang, L. Yan, Z. Yang, and Q.H. Wu. Improved document ranking in ontology-based document search engine using evidential reasoning. *IET software*, 8(1):33–41, 2014.

[3] D. Greene. *A State-of-the-art Toolkit for Document Clustering*. PhD thesis, Trinity College, 2007.

[4] M. Deepa and P. Revathy. Validation of document clustering based on purity and entropy measures. *International Journal of Advanced Research in Computer and Communication Engineering*, 1(3):147–152, 2012.

[5] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 541–544. IEEE, 2003.

[6] A. Hotho, S. Staab, and G. Stumme. Text clustering based on background knowledge. *Institute AIFB, Universität Karlsruhe*, 2003.

[7] J. Sedding and D. Kazakov. Wordnet-based text document clustering. In *proceedings of the 3rd workshop on robust methods in analysis of natural language data*, pages 104–113. Association for Computational Linguistics, 2004.

[8] L. P. Jing, L. X. Zhou, and M. K. Ng. Ontology-based distance measure for text clustering. In *Proc. of SIAM SDM workshop on text mining, Bethesda, Maryland, USA*, 2006.

[9] Y. H. Huang and X. X. Zhou. Knowledge model for electric power big data based on ontology and semantic web. *CSEE Journal of Power and Energy Systems*, 1(1):19–27, 2015.

[10] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[11] S. Vega-Pons and J. Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[12] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
[13] A. L. Gançarski, A. Doucet, and P. R. Henriques. Attribute grammar-based interactive system to retrieve information from xml documents. *IEE Proceedings-Software*, 153(2):51–60, 2006.
[14] S. Vega-Pons, J. Correa-Morris, and J. Ruiz-Shulcloper. Weighted cluster ensemble using a kernel consensus function. In *Progress in Pattern Recognition, Image Analysis and Applications*, pages 195–202. Springer, 2008.
[15] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.
[16] T. Li, C. Ding, M. Jordan, et al. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 577–582. IEEE, 2007.
[17] H. L. Luo, F. R. Jing, and X. B. Xie. Combining multiple clusterings using information theory based genetic algorithm. In *2006 International Conference on Computational Intelligence and Security*, volume 1, pages 84–89. IEEE, 2006.
[18] W3C. *Protégé*, (accessed October 3, 2013). http://protege.stanford.edu/.
[19] D. Hindle. Noun classification from predicate-argument structures. In *Proceedings of the 28th annual meeting on Association for Computational Linguistics*, pages 268–275. Association for Computational Linguistics, 1990.
[20] P. Velardi, P. Fabriani, and M. Missikoff. Using text processing techniques to automatically enrich a domain ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 270–284. ACM, 2001.
[21] Z. S. Harris. Mathematical structures of language. 1968.
[22] J. K. Cullum and R. A. Willoughby. Real symmetric matrices. In *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Vol. II Programs*, pages 11–118. Springer, 1985.
[23] D. J. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT press, 2001.
[24] F. Alabsi and R. Naoum. Comparison of selection methods and crossover operations using steady state genetic based intrusion detection system. *Journal of Emerging Trends in Computing and Information Sciences*, 3(7):1053–1058, 2012.
[25] J. Vega-Pons, S. Correa-Morris and J. Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition*, 43(8):2712–2724, 2010.
[26] J. Handl, J. Knowles and D. B. Kell. Computational cluster validation in post-genomic data analysis. *Bioinformatics*, 21(15):3201–3212, 2005.
[27] I. Korejo and S. X. Yang. Comparative study of adaptive mutation operators for genetic algorithms. *In MIC 2009: The VIII Metaheuristics International Conference*, 2, 2009.
[28] R. W. Hamming. Error detecting and error correcting codes. *Bell System technical journal*, 29(2):147–160, 1950.

Long Yan received the B.Eng. degree from the University of Liverpool (UoL), UK and Xian-jiaotong Liverpool University (XJTLU), China, respectively, in 2011, both in communication engineering and electronics. He is now pursuing the PhD degree at the University of Liverpool, UK. His research interests include asset management of power system, ontology, information retrieval, machine learning and data analysis.

W. H. Tang (M'05, SM'12) received the B.Eng. and M.Eng. degrees from Huazhong University of Science and Technology, China, in 1996 and 2000, respectively, and the Ph.D. degree from University of Liverpool,UK, in 2004. He was a Postdoctoral Research Associate from 2004 to 2006 and a Lecturer from 2006 to 2013, both in Department of Electrical Engineering and Electronics, University of Liverpool, UK. Since 2013, he is a Distinguished Professor of Thousand Talent Program for Young Outstanding Scientists in School of Electric Power Engineering at South China University of Technology, China. He has authored and co-authored over 80 research publications and also one monograph book published by Springer. His current research interests include condition monitoring and assessment for electrical equipment, wind power generation, operation risk assessment in power systems and intelligent decision support systems.

Q. H. Wu (M'91, SM'97, F'11) obtained an M.Sc.(Eng) degree in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in 1981. From 1981 to 1984, he was appointed Lecturer in Electrical Engineering in the University. He obtained a Ph.D. degree in Electrical Engineering from The Queens University of Belfast (QUB), Belfast, U.K. in 1987. He worked as a Research Fellow and subsequently a Senior Research Fellow in QUB from 1987 to 1991. He joined the Department of Mathematical Sciences, Loughborough University, Loughborough, U.K. in 1991, as a Lecturer, subsequently he was appointed Senior Lecturer. In September, 1995, he joined The University of Liverpool, Liverpool, U.K. to take up his appointment to the Chair of Electrical Engineering in the Department of Electrical Engineering and Electronics. Now he is with the School of Electric Power Engineering, South China University of Technology, Guangzhou, China, as a Distinguished Professor and the Director of Energy Research Institute of the University. Professor Wu has authored and coauthored more than 440 technical publications, including 220 journal papers, 20 book chapters and 3 research monographs published by Springer. He is a Fellow of IEEE, Fellow of IET, Chartered Engineer and Fellow of InstMC. His research interests include nonlinear adaptive control, mathematical morphology, evolutionary computation, power quality and power system control and operation.

J. S. Smith received the B.Eng. (Hons) and Ph.D. degrees in engineering from The University of Liverpool, Liverpool, U.K., in 1984 and 1990, respectively. Between 1984 and 1988, he was conducting research on image processing and robotic systems with the Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, U.K. He has worked as a Lecturer, Senior Lecturer, and Reader in the same department since 1988. He has held a Professorship since 2006 in Electrical Engineering at The University of Liverpool. His research interests include automated welding, robotics, vision systems, adaptive control, digital systems, FPGAs and embedded computer systems.